Handling High Parameter Dimensionality in Reinforcement Learning with Dynamic Motor Primitives

Adrià Colomé, Guillem Alenyà and Carme Torras

***Abstract*— Dynamic Motor Primitives (DMP) are nowadays widely used as movement parametrization for learning trajectories, because of their linearity in the parameters, rescalation robustness and continuity. However, when learning a movement with DMP, where a set of gaussians distributed along the trajectory is used to approximate an acceleration excitation function, a very large number of gaussian approximations need to be performed. Adding them up for all joints yields too many parameters to be explored, thus requiring a prohibitive number of experiments/simulations to converge to a solution with an optimal (locally or globally) reward.**

We propose here two strategies to reduce this dimensionality: the first is to explore only the most significant directions in the parameter space, and the second is to add a reduced second set of gaussians that should only optimize the trajectory after fixing the gaussians that approximate the demonstrated movement.

I. Dynamic Motor Primitives

Dynamic Motor Primitives [1] characterize a movement by means of a dynamical system, using a position error, a velocity term and an excitation function for obtaining the acceleration profile generating the movement:

$$\begin{aligned}\dot{z}/\tau &= \alpha_z\left(\beta_z\left(g-y\right)-z\right)+\boldsymbol{\theta}^T\mathbf{g}(t)\\ \dot{y}/\tau &= z,\end{aligned}$$

where y is one joint position, g the goal/ending joint position, τ a time constant and z a rescaled velocity. In addition, $\boldsymbol{\theta}$ is the parameter vector used to learn an initial move, applied to a set of basis functions $\mathbf{g}(t)$, defined as:

$$\begin{aligned}g_i(t) &= \frac{\phi_i(x(t))}{\sum_j \phi_j(x(t))}x(t)\\ \phi_i\left(x(t)\right) &= exp\left(-0.5(x(t)-c_i)^2/d_i\right),\end{aligned}$$

c_i, and d_i representing the fixed center and width of the *i*th gaussian used. Also, x is a transformation of time verifying $\dot{x}=-\alpha_x x/\tau$.

With this movement representation, the robot can be taught a demonstration movement, to obtain the weights and gaussians of a $f_{demo}(t)=\boldsymbol{\theta}^T\mathbf{g}(t)$. However, each robot joint will have its own DMP, which results in a very large number of gaussians to have a good and chattering-free approximation of the taught movement, easily having over 100 parameters for a 7 degrees-of-freedom (dof) arm. This motivated us to look for new exploration strategies to be able to learn movements in unstructured environments which cannot be simulated, and where thus it is critical to reduce the number of experiments.

This work is partially funded by EU Project IntellAct (FP7-269959) and by the Spanish Ministry of Science and Innovation under project PAU+ DPI2011-27510.

The authors are with Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens Artigas 4-6, 08028 Barcelona, Spain. E-mails: [acolome,galenya,torras]@iri.upc.edu

II. Exploring in Significant Directions

A first option is to explore in the direction of those parameter vectors that have most influence on the trajectory. This can be done by computing (only once), the matrix:

$$\mathbf{W}=\begin{bmatrix}\phi_1\left(x(t_0)\right) & \ldots & \phi_M\left(x(t_0)\right)\\ \ldots & & \ldots\\ \phi_1\left(x(t_N)\right) & \ldots & \phi_M\left(x(t_N)\right)\end{bmatrix},$$

M being the number of gaussians for each joint of the robot and N the number of timesteps in the trajectory. This matrix $\mathbf{W}$ will have dimension $N\times M$, with $N>>M$. If we compute its Singular Value Decomposition, we get its eigenvectors $\mathbf{v}_1,...,\mathbf{v}_M$ in the parameter space (dimension M), with gains associated to the singular values obtained $\sigma_1>...>\sigma_M$. Then, instead of exploring each DMP parameter θ_i, we can take the $K<M$ most relevant directions $\mathbf{v}_1,...,\mathbf{v}_K$ and, for each exploration repetition of the movement, use an updated parameter computed as $\boldsymbol{\theta}_e=\boldsymbol{\theta}+\epsilon_1\mathbf{v}_1+...+\epsilon_K\mathbf{v}_K$, with ϵ_s following a normal distribution with a predefined exploration variance.

III. Dual-Layer excitation functions

Another strategy to tackle high-dimensionality in the exploration is to create a dual-layer excitation function, in the sense that, being the DMP main equation

$$\frac{1}{\tau}\dot{z}=\alpha_z\left(\beta_z\left(g-y\right)-z\right)+\mathbf{F}(x(t)),$$

one can take $\mathbf{F}(\mathbf{x(t)})=\boldsymbol{\theta_0}^T\mathbf{g_0}(t)+\boldsymbol{\theta_e}^T\mathbf{g_e}(t)$, with $\boldsymbol{\theta_0}^T$ and $\mathbf{g_0}(t)$ the weights and gaussians of the excitation function learned with the demonstrated move, and $\boldsymbol{\theta_e}^T$ and $\mathbf{g_e}(t)$ a reduced set of gaussians that does not need to approximate the learned movement but just to optimize the trajectory and thus it is less constrained and can have wider kernels in order to avoid oscillations coming from random exploration. $\boldsymbol{\theta_e}^T$ can be initialized to zero and be updated as in any learning algorithm in literature, such as policy gradients [2] or path integral approaches [3]. Experimental results using both strategies will be presented at the workshop

References

[1] A. J. Ijspeert, J. Nakanishi and S. Schaal. "Movement imitation with nonlinear dyamical systems in humanoid robots." *Proc. IEEE ICRA*, pp 1398-1403, 2002.

[2] J. Peters, S. Schaal. "Policy gradient methods for robotics ." *Proc. IEEE/RSJ IROS*, pp 2219-2225, 2006.

[3] F. Stulp, E. A. Theodorou, S. Schaal. "Reinforcement learning with sequences of motion primitives for robust manipulation" *IEEE Transactions on robotics*, vol 28, no 6, 2012.